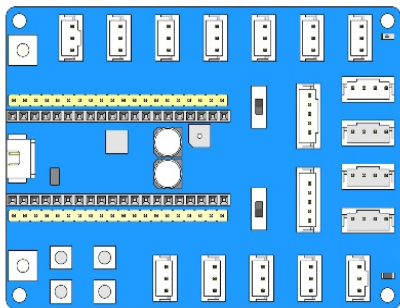




Lesson 3- Testing LEDS,SW2812,buzzer on Pico Expansion Board

In this lesson, we will use Raspberry Pi Pico to testing LEDS,SW2812,buzzer on Pico Expansion Board

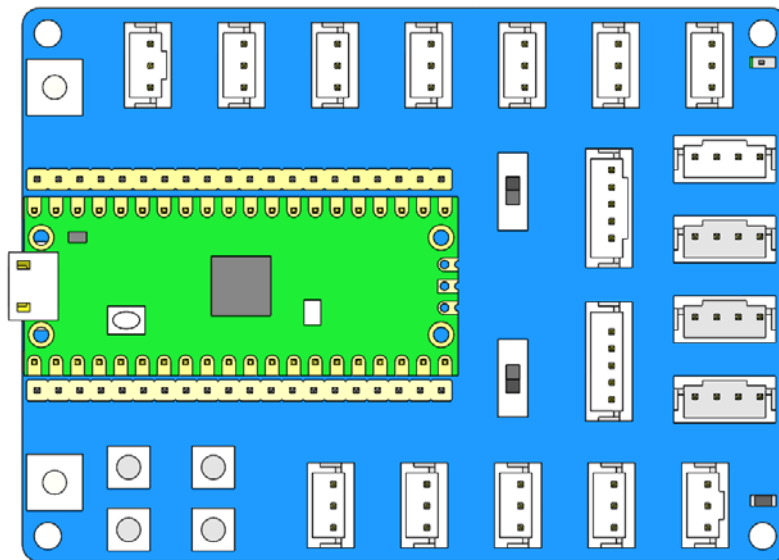
1.Component List

Component	QTY	Picture
Pico Expansion Board	1	
Raspberry Pi Pico	1	
USB Cable	1	

2. Circuit

Put assemble the Raspberry Pico on the Pico Expansion board

Hardware connection:



3. Upload Arduino code to Pico and testing

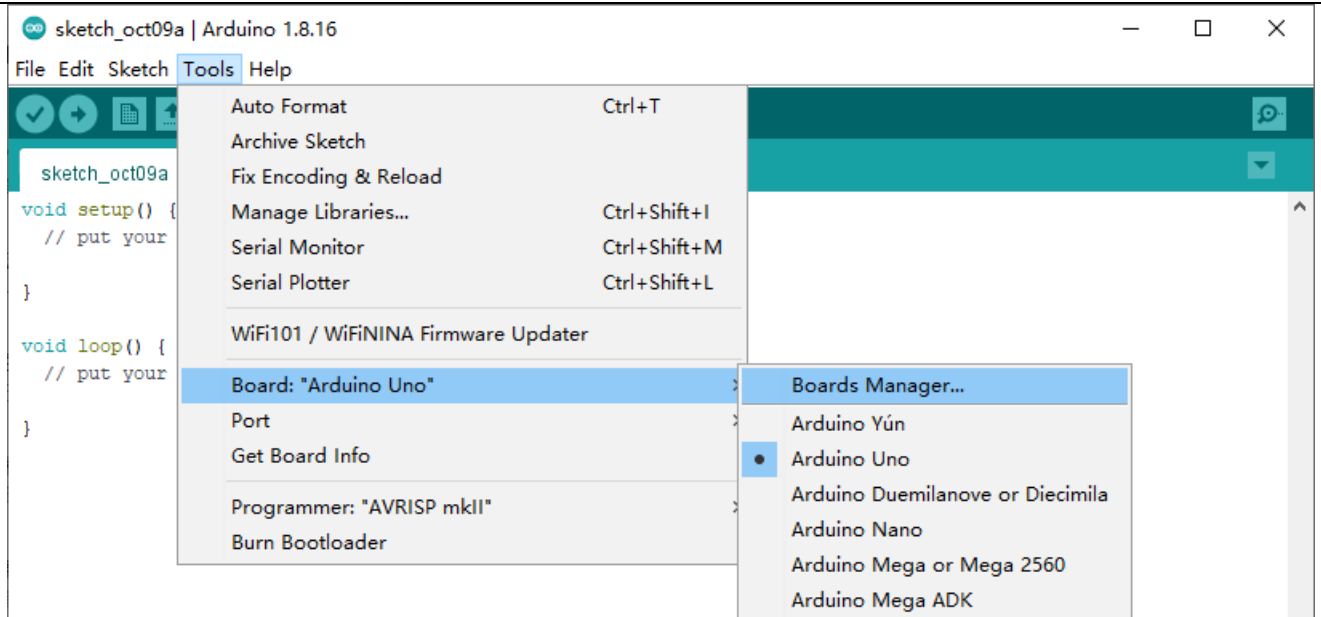
Configuring the Operating Environment For Arduino

Please refer to the document “[3 Configuring the Operating Environment For Arduino](#)” which saved in ‘[Pico_Extension_Board Kit Tutorial](#)’

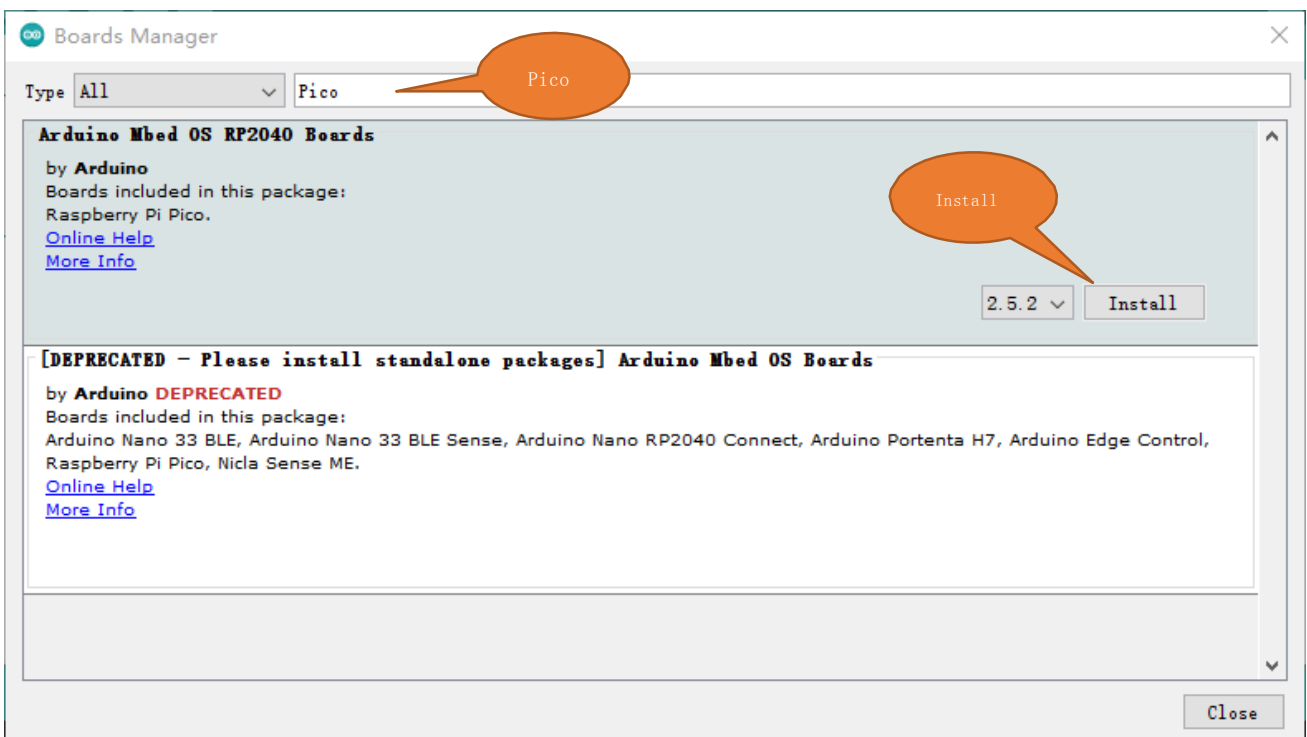
3.1 Installation of Development Board Support Package for Pico

3.1.1 Make sure your network is of good connection.

3.1.2 Open Arduino IDE. Click Tools>**Board**>**Boards Manager...**on the menu bar.



3.1.3 Enter Pico in the searching box, select "Arduino Mbed OS RP2040 Boards" and click on Install.



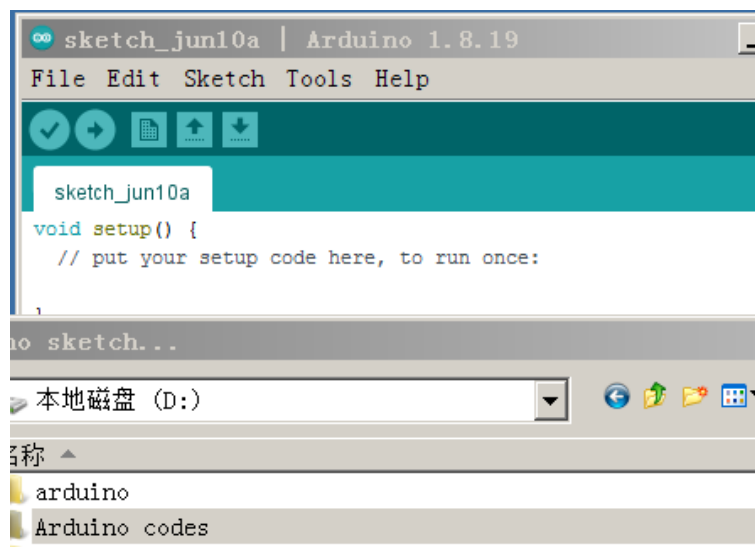
3.1.4 Click Yes in the pop-up "dpinst-amd64.exe" installation window. (Without it, you will fail to communicate with Arduino.) Thus far, we have finished installing the development support package.

3.3 Uploading testing Arduino code to Pico

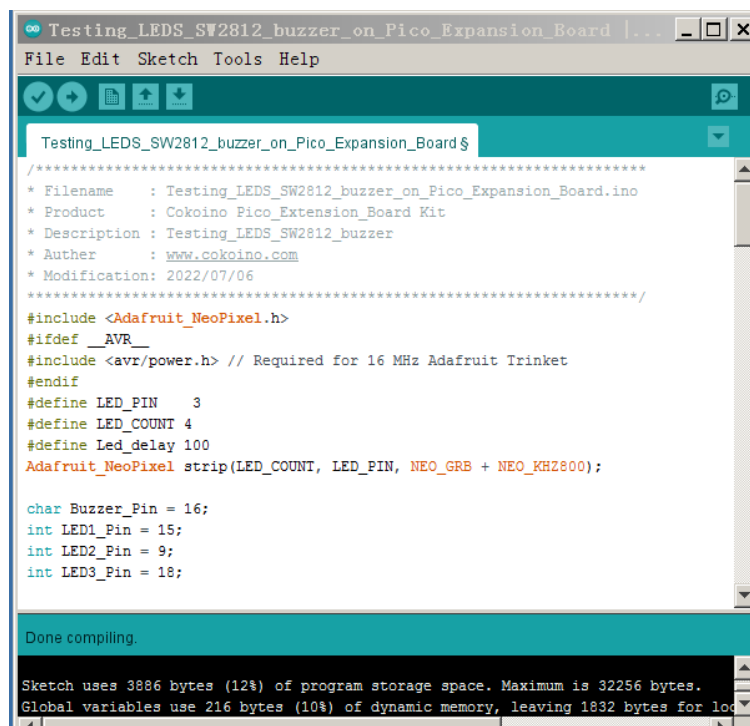
Codes used in this lesson are saved in "Pico_Extension_Board Kit Tutorial\Codes\Arduino_Codes". You can move the codes to any location. For example, we save the codes in Disk(D) with the path of "D:/ Arduino codes".

Testing LEDS,SW2812,buzzer on Pico Expansion Board

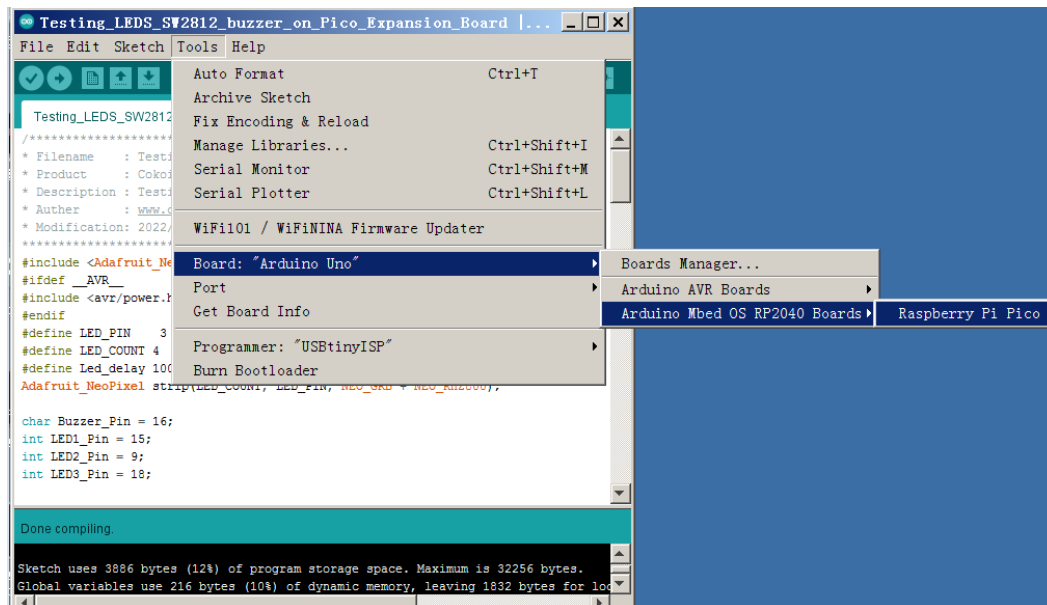
Open "Arduino" IDE, click "File"--"Open"--"D:"--"Arduino codes".



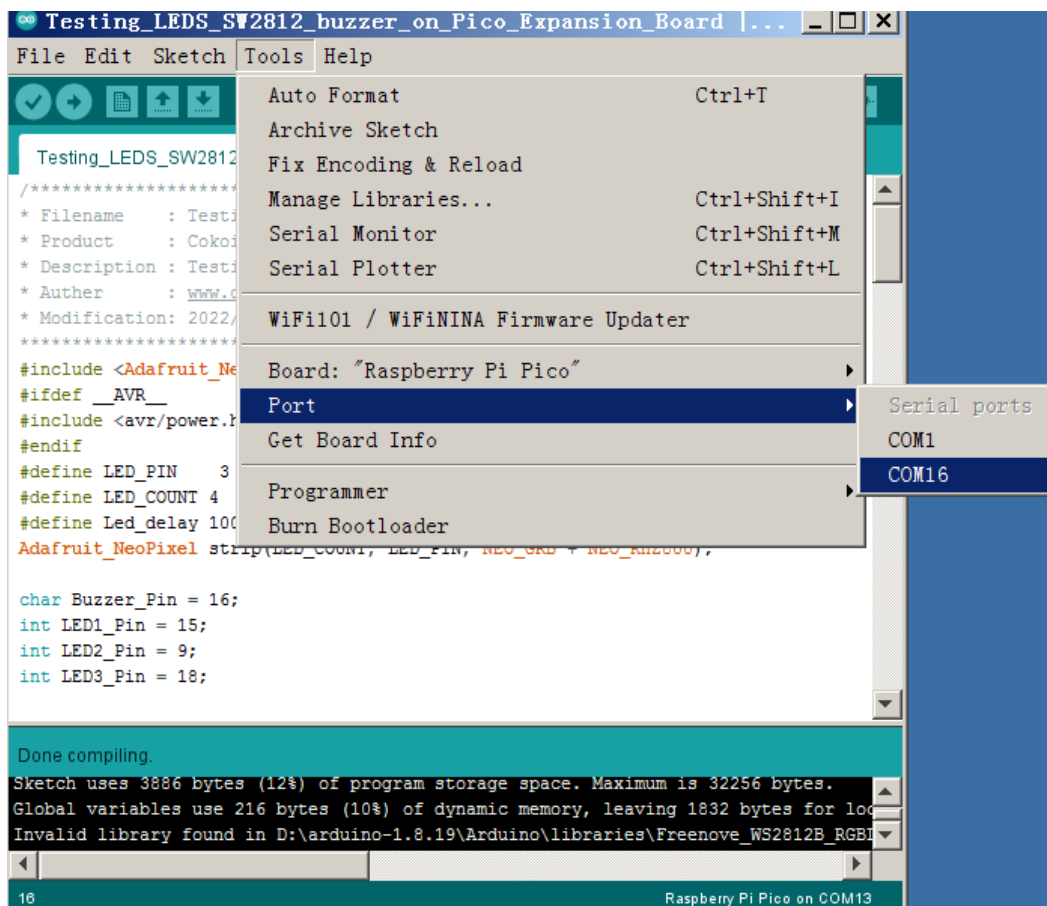
Expand folder "Testing_LEDS_SW2812_buzzer_on_Pico_Expansion_Board" and double click "Testing_LEDS_SW2812_buzzer_on_Pico_Expansion_Board.ino" open it. As shown in the illustration below.



Make sure Raspberry Pi Pico has been connected with the computer by usb cable. Then click “Tools”-“Board”-“Arduino Mbed OS RP2040 Boards”-“Raspberry Pi Pico” As shown in the illustration below.



Click “Tools”-“Port” to select the Serial ports for Raspberry Pi Pico .As shown in the illustration below.



Upload the code to Raspberry Pi Pico as shown in the illustration below.



```
Testing_LEDS_SW2812_buzzer_on_Pico_Expansion_Board
File Edit Sketch Tools Help

Testing_LEDS_SW2812_buzzer_on_Pico_Expansion_Board
/*****
* Filename      : Testing_LEDS_SW2812_buzzer_on_Pico_Expansion_Board.ino
* Product       : Cokoino Pico_Extension_Board Kit
* Description    : Testing_LEDS_SW2812_buzzer
* Author        : www.cokoino.com
* Modification: 2022/07/06
*****/

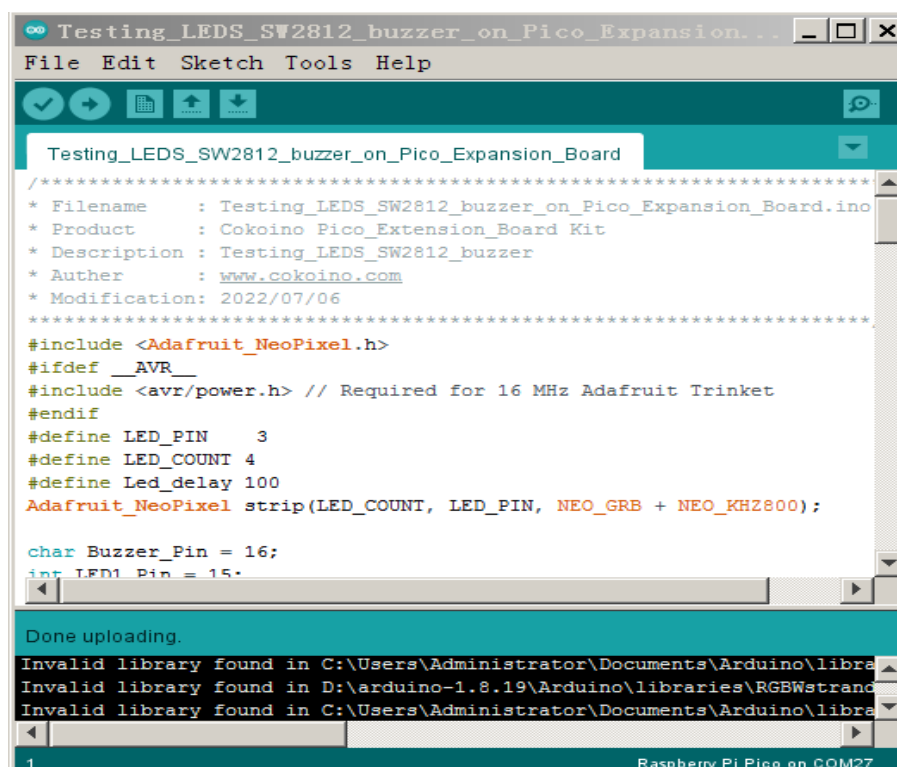
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif
#define LED_PIN    3
#define LED_COUNT  4
#define Led_delay  100
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

char Buzzer_Pin = 16;
int LED1_Pin = 15;
int LED2_Pin = 9;
int LED3_Pin = 18;

Compiling sketch...

16 Raspberry Pi Pico on COM16
```

Upload succeeded as shown in the illustration below.



```
Testing_LEDS_SW2812_buzzer_on_Pico_Expansion_Board
File Edit Sketch Tools Help

Testing_LEDS_SW2812_buzzer_on_Pico_Expansion_Board
/*****
* Filename      : Testing_LEDS_SW2812_buzzer_on_Pico_Expansion_Board.ino
* Product       : Cokoino Pico_Extension_Board Kit
* Description    : Testing_LEDS_SW2812_buzzer
* Author        : www.cokoino.com
* Modification: 2022/07/06
*****/

#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif
#define LED_PIN    3
#define LED_COUNT  4
#define Led_delay  100
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

char Buzzer_Pin = 16;
int LED1_Pin = 15;

Done uploading.
Invalid library found in C:\Users\Administrator\Documents\Arduino\libra
Invalid library found in D:\arduino-1.8.19\Arduino\libraries\RGBWstrand
Invalid library found in C:\Users\Administrator\Documents\Arduino\libra

1 Raspberry Pi Pico on COM27
```

Notes:

1. Be sure to keep pressing the “BOOTSEL” button before powering the Pico, otherwise the firmware will not download successfully at the first time.
2. At the first time you use Arduino to upload sketch for Pico, you don't need to select port. After that, each time before uploading sketch, please check whether the port has been selected; otherwise, the downloading may fail.
3. Sometimes when using, Pico may lose firmware due to the code and fail to work. At this point, you can upload firmware for Pico as mentioned above

4. Testing results

After upload the testing code to RaspberryPico, please re-plug the usb cable in the Pico.

You will see that the 4 sw2812 LEDs on the Pico expansion board flashing in red-green-blue-white order, GP9/GP15/GP18 red LEDs on the Pico expansion board flashing in 1second cycle, the buzzer on the Pico expansion board circulates a beeping sound

Code as below:

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h> // Required for 16 MHz Adafruit Trinket
#endif
#define LED_PIN 3 // Which pin on the Arduino is connected to the NeoPixels?
#define LED_COUNT 4 // Which pin on the Arduino is connected to the NeoPixels?
#define Led_delay 100
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);

char Buzzer_Pin = 16;
int LED1_Pin = 15;
int LED2_Pin = 9;
int LED3_Pin = 18;

void setup()
{
  #if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
    clock_prescale_set(clock_div_1);
  #endif
  // END of Trinket-specific code.
  strip.begin(); // INITIALIZE NeoPixel strip object (REQUIRED)
  strip.show(); // Turn OFF all pixels ASAP
  strip.setBrightness(50); // Set BRIGHTNESS to about 1/5 (max = 255)
  Serial.begin(9600);
}

void loop()
{
  SW2812_Test();
  Buzz_Test();
  LED_Test();
}

void Buzz_Test()
{
  for (char i = 0; i < 80; i++)
  {
    digitalWrite(Buzzer_Pin, HIGH);
```

```
    delay(1);
    digitalWrite(Buzzer_Pin,LOW);
    delay(1);
  }
  for (char i = 0; i < 100; i++)
  {
    digitalWrite(Buzzer_Pin,HIGH);
    delay(2);
    digitalWrite(Buzzer_Pin,LOW);
    delay(2);
  }
}
void LED_Test()
{
  pinMode( LED1_Pin,OUTPUT);
  pinMode( LED2_Pin,OUTPUT);
  pinMode( LED3_Pin,OUTPUT);
  digitalWrite(LED1_Pin,HIGH);
  digitalWrite(LED2_Pin,HIGH);
  digitalWrite(LED3_Pin,HIGH);
  delay(1000);
  digitalWrite(LED1_Pin,LOW);
  digitalWrite(LED2_Pin,LOW);
  digitalWrite(LED3_Pin,LOW);
}
void SW2812_Test()
{
  colorWipe(strip.Color(255, 0, 0), Led_delay); // Red
  colorWipe(strip.Color( 0, 255, 0), Led_delay); // Green
  colorWipe(strip.Color( 0, 0, 255), Led_delay); // Blue
  colorWipe(strip.Color( 255,255, 255), Led_delay); // White
  //rainbow(10); // Flowing rainbow cycle along the whole strip
  //theaterChaseRainbow(50); // Rainbow-enhanced theaterChase variant
}

void colorWipe(uint32_t color, int wait) {
  for(int i=0; i<strip.numPixels(); i++) { // For each pixel in strip...
    strip.setPixelColor(i, color); // Set pixel's color (in RAM)
    strip.show(); // Update strip to match
    delay(wait); // Pause for a moment
  }
}

void theaterChase(uint32_t color, int wait)
{
  for(int a=0; a<10; a++) { // Repeat 10 times...
    for(int b=0; b<3; b++) { // 'b' counts from 0 to 2...
      strip.clear(); // Set all pixels in RAM to 0 (off)
      // 'c' counts up from 'b' to end of strip in steps of 3...
      for(int c=b; c<strip.numPixels(); c += 3) {
        strip.setPixelColor(c, color); // Set pixel 'c' to value 'color'
      }
      strip.show(); // Update strip with new contents
      delay(wait); // Pause for a moment
    }
  }
}

void rainbow(int wait)
{
  for(long firstPixelHue = 0; firstPixelHue < 5*65536; firstPixelHue += 256)
  {
    strip.rainbow(firstPixelHue);
    strip.show(); // Update strip with new contents
    delay(wait); // Pause for a moment
  }
}
```



```
}  
  
void theaterChaseRainbow(int wait)  
{  
  int firstPixelHue = 0; // First pixel starts at red (hue 0)  
  for(int a=0; a<30; a++) { // Repeat 30 times...  
    for(int b=0; b<3; b++) { // 'b' counts from 0 to 2...  
      strip.clear(); // Set all pixels in RAM to 0 (off)  
      for(int c=b; c<strip.numPixels(); c += 3)  
      {  
        int hue = firstPixelHue + c * 65536L / strip.numPixels();  
        uint32_t color = strip.gamma32(strip.ColorHSV(hue)); // hue -> RGB  
        strip.setPixelColor(c, color); // Set pixel 'c' to value 'color'  
      }  
      strip.show(); // Update strip with new contents  
      delay(wait); // Pause for a moment  
      firstPixelHue += 65536 / 90; // One cycle of color wheel over 90 frames  
    }  
  }  
}
```

What's Next?

THANK YOU for participating in this learning experience!

If you find errors, omissions or you have suggestions and/or questions about the Tutorial or component contents of this Kit, please feel free to contact us: cokoino@outlook.com

We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website. We will continue to launch fun, cost-effective, innovative and exciting products.

<http://cokoino.com/>

Thank you again for choosing Cokoino products.